ГУАП

КАФЕДРА №43

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

Ассистент		А.А. Фоменкова
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №7

ОРГАНИЗАЦИЯ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ ПРИ РЕШЕНИИ ЧИСЛЕННЫХ ЗАДАЧ

по курсу: ИНФОРМАТИКА

Цель работы

Научиться организовывать текстовый интерфейс пользователя. Получение навыков написания программ для решения численных задач с текстовым интерфейсом пользователя

№ варианта	Задание	Функция для демонстрационного примера
5	г	0.05sint

1. Основное меню

```
clear all;
clc;
userStr = '';
time = -1;
vecX = []; vecY = []; vecZ = [];
inf();
menuState = 'M';
nextMenuState = 'M';
while (true)
        switch menuState
                case 'M' %Главное меню
                        disp('Главное меню glavMenu');
                        disp('1 - Демонстрация работы программы');
                        disp('2 - Расчет угла крена');
                        disp('3 - Информация о программе и авторе');
                        disp('4 - Выход');
                        menuState = Menu4;
                        clc;
                        continue;
                case 'S' %Подменю 2
                        disp('Меню расчетов');
                        disp('1 - Задать новые значения');
                        disp('2 - Возврат в главное меню');
                        disp('3 - Завершение работы');
                        menuState =  Menu3;
                        menuState = sprintf('S%s', menuState);
                        clc;
                        continue;
                case '1' %Демонстрация
                        disp('Демонстрация программы');
                        [vecXT, vecYT, vecZT] = countGlavMenu('0.05*sin(t^2)', 10);
                        printTable2D(vecXT, vecYT);
                        disp('Для возврата в главное меню нажмите любую клавишу')
                        pause;
                        clc;
                        nextMenuState = 'M';
                case '2' %Расчет
                        if(isequal(userStr, '') || time < -1)
                                menuState = 'I';%Принудительно назначает стадию ввода
                                nextMenuState = '2';
                                continue;
                        end
                        printTable2D(vecX, vecY);
                        %Расчет успешен.
                        menuState = 'S';
                        nextMenuState = '2';
                        continue;
                case '3' %Информация
                        inf();
                        nextMenuState = 'M';
                case '4' %Выход
                        clc;
                        clear all;
                        close();
                        break;
                case 'S1' %Ретранслятор с подменю 2 до меню выбора
                        menuState = 'I';
                        continue;
                case 'S2' %Ретранслятор с подменю 2 до главного меню
```

```
                nextMenuState = 'M';
            case 'S3' %Ретранслятор с подменю 2 до меню выхода
                nextMenuState = '4';
            case 'I' % Запрашивает данные
                while(true)
                    userStr = input('Введите f(t) или вектор дискертных значений: ', 's');
                    userStr = strrep(userStr, ',', '.');
                    if(~isempty(str2num(userStr)))
                        sizeMatrix = size(str2num(userStr));
                        if(sizeMatrix(1) > 1)
                            disp('Произошла ошибка ввода, повторите попытку.');
                            continue;
                        end
                        disp('Введите интервал времени между измерениями в векторе.');
                        break;
                    else
                        disp('Введите время измерения.');
                        break;
                    end
                end
                time = inputTime();
                disp('Начинается предварительный расчет, ожидайте.');
                [vecX, vecY, vecZ] = GlavMenu(userStr, time);
                clc;
            otherwise
                clc;
                clear all;
                close();
                disp('Критическая ошибка. Доложите об этом!');
                break;
        end
        menuState = nextMenuState;
end
clear all;
```


2. Подпрограмма

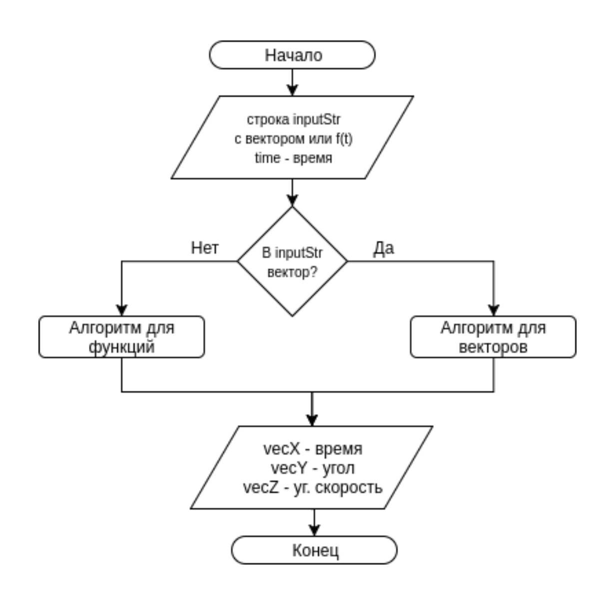
```
function [vecX vecY vecZ] = GlavMenu(inputStr, time)
%count GlavMenu это угол
% Syntax:[vecX vecY vecZ] = GlavMenu(inputStr, time)
% vecX - вектор значения времени t
% vecY - вектор значения угла
% vecZ - вектора значения угловой скорости
% Если во входной строке вектор, то time - время между измерениями вектора
% Если во входной строке f(t), то time время, до какого момента считается интеграл
    if ( isempty( str2num(inputStr) ) )
        %В строке не вектор, используем алгоритм для функции
        iMax = time/0.1 + 1;
        vecX = 0:0.1:time;
        vecY(1:iMax) = 0;
        vecZ(1:iMax) = 0;

        for i = 1:iMax
            %Расчитываем угловую скорость
            t = vecX(i);
            vecZ(i) = eval(inputStr);
            %Расчитываем угол
            y = 0;
            x = vecX(i);
            for t = 0:0.01:x
                y = y + eval(inputStr);
            end
            vecY(i) = y*0.01;
        end
        return;
    else
        % Алгоритм для векторов
        vector = str2num(inputStr);

        iMax = numel(vector) + 1;
```

```
            vecX = 0:time:((iMax-1)*time);
            vecY(1:iMax) = 0;
            vecZ(2:iMax) = vector;
            for i = 1:iMax
                vecY(i) = sumVector(vecZ, 1, i) * time; % Это интеграл
            end
            return;
        end
end
```

2. Блок схема



3 Вывод программы

```
a =
     0    0.1000    0.2000    0.3000    0.4000    0.5000    0.6000    0.7000    0.8000    0.9000    1.0000
b =
     0    0.0055    0.0209    0.0461    0.0809    0.1248    0.1775    0.2384    0.3069    0.3823    0.4639
c =
     0    0.0998    0.1987    0.2955    0.3894    0.4794    0.5646    0.6442    0.7174    0.7833    0.8415
```

4. Функция Menu

```
function [menu] =  menu
%inMenu - Запрашивает целочисленое число от 1 до input
%
% Syntax: [menu] = inMenu(

        while (true)

            strInput = input('Введите пункт меню: ', 's');
            strInput = strrep(strInput, ',', '.'); %Замена запятых на точки для корректной работы с десятичными дробями
            in = str2double(strInput);
            if(isnan(in))
                disp('Ошибка ввода.');
                continue;
            end
            onlyFractPart = in - floor(in);
            if(onlyFractPart == 0) % Проверка на целочисленость
                if(in >= 1 && in <= menus)
                    menu = strInput;
                    return;
                end
            end
            disp('Ошибка ввода.');
        end
end
```

4.2 Вывод

```
>> inputMenu(3)
Введите пункт меню: 5
```

Ошибка ввода.
Введите пункт меню: S
Ошибка ввода.
Введите пункт меню: 2
ans =
    '2'

4. Функция запрашивает номер меню

```
function [menu] = inputmenu
%inputMenu - Запрашивает целочисленое число от 1 до input
% Syntax: [menu] = inputMenu(menus)
	while (true)
		strInput = input('Введите пункт меню: ', 's');
		strInput = strrep(strInput, ',', '.'); %Замена запятых на точки для корректной работы с десятичными дробями
		in = str2double(strInput);
		if(isnan(in))
			disp('Ошибка ввода.');
			continue;
		end
		onlyFractPart = in - floor(in);
		if(onlyFractPart == 0) % Проверка на целочисленость
			if(in >= 1 && in <= menus)
				menu = strInput;
				return;
			end
		end
		disp('Ошибка ввода.');
	end
end
```


4.2 Вывод
>> inputMenu(3)
Введите пункт меню: 5
Ошибка ввода.
Введите пункт меню: S
Ошибка ввода.
Введите пункт меню: 2
ans =
    '2'

5. Функция inputTime запрашивает положительное число

```
function [out] = inputTime()
%inputTime - Запрашивает положительное число у пользователя
%
% Syntax: [out] = inputTime()

	while(true) %Цикл
		in = input('Введите положительное число: ', 's');
		in = strrep(in, ',', '.');
		digit = str2double(in);
		if (isnan(digit) || digit < 0) %Если не число запрашиваем опять
			disp('Неверный ввод. Повторите попытку');
			continue;
		else
			out = digit;
		return;
		end
	end
end
```

5.Вывод

>> inputTime()
Введите положительное число: s
Неверный ввод. Повторите попытку
Введите положительное число: -3
Неверный ввод. Повторите попытку
Введите положительное число: 9.9999
ans =
    9.9999

6. Листинг вектора и сложения

```
function [S] = sumVector(vector, ibegin, iend)
% SumVector - Складывает все элементы вектора от ibegin до iend
% Syntax: S = sumVector(vector, ibegin, iend)
S = 0;
for i = ibegin:iend
            S = S + vector(i);
end
```

6. Вывод

```
>> vec = [90 -90 30 -30 9 0 16 32 -0];
>> sumVector(vec, 3, 6)
ans =
    9
```